C.) Amendments to the Claims

1. (Previously Presented) A network data processor system comprising a plurality of data packet processors coupled through a data switch fabric between network connection processors, wherein said data packet processors perform a data processing function over data contained within predetermined data packets, wherein said network connection processors include network interfaces coupleable to external data transmission networks and wherein said network connection processors provide for the selective routing of said predetermined data packets through said data switch fabric to load balance the processing of said predetermined data packets by said plurality of data packet processors.

2. (Currently Amended) A network data packet processor system providing for the transfer of packets between first and second networks, said network data packet processor system comprising:

a) a data packet switch including pluralities of first and second data ports coupled together to provide for the transfer of network data packets between respective first and second data ports;

b) a plurality of data protocol processors coupled to a like plurality of said first data ports of said data packet switch, each data protocol processor being coupled to a respective first data port through a bidirectional packet transfer interface and including a protocol processing engine providing for the selective conversion of data contained within a predetermined network data packet; and

[[b)]] c) input and output data transfer processors coupled to respective second data ports of said data packet switch, wherein said input data transfer processor selectively routes network data packets from said first network to said plurality of data protocol processors and said output data transfer processor routes network data packets from said plurality of protocol processors to said second network, and wherein said input data transfer processor balances the load of individual network data packets routed to said plurality of data protocol processors.

3. (Original) A network gateway processor comprising:

a) a switch providing data routing between input, output, and processing ports;

b) an array of protocol processors coupled to respective processing ports, each said protocol processor providing for the conversion of network data packets from a first form to a second form;

c) an input processor coupled between a first network and said input port, said input processor providing for the load balanced allocation of network data packets received from said first network to said array of protocol processors; and

d) an output processor coupled between a second network and said output port, wherein said array of protocol processors provide network data packets of said second form to said output processor for transfer to said second network.

4. (Original) The network gateway processor of Claim 3 wherein said input processor selectively associates conversion control data with network data packets provided to said array of protocol processors.

5. (Original) The network gateway processor of Claim 4 wherein said conversion control data is provided with each network data packet provided to said array of protocol processors.

6. (Original) The network gateway processor of Claim 5 wherein each said protocol processor includes a data form conversion engine and wherein operation of said data form conversion engine is defined by predetermined parameters identified by said conversion control data and wherein said predetermined parameters are applied to said data form conversion engine with respect to a corresponding network data packet.

7. (Original) The network gateway processor of Claim 6 wherein said data form conversion engine includes an encryption engine.

8. (Original) A method of operating a network gateway coupleable between first and second networks to implement a compute intensive data processing function on network data packets transferred between said first and second networks, said method comprising:

a) receiving, by a first processor coupleable to said first network, network data packets;

b) selecting, from said received network data packets, predetermined network packets for routing through said network gateway;

c) selectively distributing said predetermined network data packets to a plurality of second processors so as to enable utilization of the aggregate performance of said second processors in performing said compute intensive data processing function;

d) processing, asynchronously, said predetermined network data packets as distributed by said plurality of second processors to convert each of said predetermined network data packets in accordance with said compute intensive data processing function to provide converted network data packets;

e) collecting, by a third processor coupleable to said second network, said converted network data packets; and

f) transferring said converted network data packets to said second network.

9. (Currently Amended) The method of Claim 8 wherein said compute intensive data processing function <u>is</u> one or a combination of functions selected from a group consisting of data encryption, decryption, compression, decompression, and protocol translation.

10. (Original) The method of Claim 8 wherein said compute intensive data processing function is dependant on configuration parameters and wherein said method further comprising the steps of;

a) obtaining said configuration parameters; and

b) applying said configuration parameters, within said step of processing, to control the conversion of each of said predetermined network data packets.

11. (Original) The method of Claim 10 wherein said step of obtaining includes negotiating, by a fourth processor, a set of configuration parameters for a

predetermined logical connection established through said network gateway between said first and second networks and wherein said step of applying includes selecting said set of configuration parameters with respect to a predetermined network packet associated with said predetermined logical connection.

12. (Original) The method of Claim 11 further comprising the steps of:

a) distributing, by said fourth processor to said first processor, said set of configuration parameters; and

b) associating, by said first processor, said set of configuration parameters with said predetermined network packet such that said set of configuration parameters is passed, in combination with said predetermine network packet by said step of selectively distributing, to a predetermined one of said plurality of second processors.

13. (Original) The method of Claim 11 further comprising the steps of:

a) distributing, by said fourth processor to said second processors, said set of configuration parameters; and

b) associating, by a predetermined one of said second processors, said set of configuration parameters with said predetermined network packet as passed by said step of selectively distributing, to said predetermined one of said plurality of second processors.

14. (Currently Amended) The method of Claim 12 wherein said compute intensive data processing function is one or a combination of functions selected from a group consisting of data encryption, decryption, compression, decompression, and protocol translation.

15. (Original) The method of Claim 14 wherein said compute intensive data processing function implements a conversion between an IP protocol and an IPsec protocol.

16. (Original) A method of performing compute intensive protocol transformation functions on network data, said method comprising the steps of:

a) receiving, through a first network connection, select network data packets for protocol transformation;

b) distributing said select network data packets to a plurality of protocol transformation processors;

c) converting, by said plurality of protocol transformation processors, said select network data packets in accordance with said protocol transformation to provide converted network data packets;

d) collecting said converted network data packets from said plurality of protocol transformation processors; and

e) sending said converted network data packets through a second network connection.

17. (Original) The method of Claim 16 wherein said step of converting includes determining for each select network data packet a corresponding set of parameters for use in performing said protocol transformation, said method further comprising the step of dynamically developing said corresponding set of parameters.

18. (Original) The method of Claim 17 wherein said corresponding set of parameters is dynamically developed for a logical connection established between said first and second network connections.

19. (Original) The method of Claim 18 wherein said protocol transformation is an implementation of a secure IP protocol.

20. (Original) The method of Claim 19 wherein said logical connection is a virtual private network and wherein said protocol transformation implements a conversion between an IP protocol and an IPsec protocol.

21. (Original) A network gateway supporting a compute intensive protocol processing function for transferred data packets, said network gateway comprising:

a) a switch fabric implementing programmable channel transfer of data between first, second, and third fabric interface ports;

b) an ingress processor coupleable to a first network and coupled to said first fabric interface port to transfer data packets defined in accordance with a first protocol format from said first network to said switch fabric;

c) an egress processor coupleable to a second network and coupled to said second fabric interface port to transfer data packets defined in accordance with a second protocol format from said switch fabric to said second network; and

d) a parallel array of protocol processors coupled to respective instances of said third interface port of said switch fabric to receive data packets from said ingress processor and send data packets to said egress processor, said parallel array of protocol processors implementing a compute intensive network packet transformation function between said first and second protocol formats for data packets passed through said parallel array of protocol processors;

whereby the aggregate throughput performance of said parallel array of protocol processors directly supports the throughput performance of said ingress processor.

22. (Original) The network gateway of Claim 21 wherein said ingress processor determines the distribution of received data packets to the individual protocol processors of said parallel array.

23. (Original) The network gateway of Claim 22 further comprising a control processor coupled within said network gateway to communicate protocol processing parameters to said parallel array of protocol processors to selectively control the execution of said compute intensive network packet transformation function by the individual protocol processors of said parallel array.

24. (Original) The network gateway of Claim 23 wherein said protocol processing parameters are transferred by said control processor to said ingress processor and wherein said ingress processor selectively associates said protocol processing parameters with data packets transferred to said parallel array of protocol processors.

25. (Original) The network gateway of Claim 23 wherein said compute intensive network packet transformation function implements a secure IP protocol,

wherein said protocol processing parameters are dynamically negotiated by said control processor according to said secure IP protocol.

26. (Original) The network gateway of Claim 25 wherein said control processor is coupled through said switch fabric to transfer said protocol processing parameters to a data table stored by said ingress processor, wherein said ingress processor dynamically attaches headers selectively containing said protocol processing parameters to data packets prior to transfer to said parallel array of protocol processors, the selection of said protocol processing parameters being dependent on information contained in respective data packets.

27. (Original) The network gateway of Claim 26 wherein each protocol processor of said parallel array includes a data table, wherein said control processor is coupled through said switch fabric to transfer said protocol processing parameters to each said data table, and wherein each protocol processor of said parallel array determines from received data packet select parameters of said protocol processing parameters to use in said compute intensive network packet transformation function as implemented by respective ones of said parallel array.